

Algorithms for the selection of the active sensors in distributed tracking: comparison between Frisbee and GNS methods

A. Capponi, C. Pilotto
California Institute of
Technology
Computer Science
1200 E. California
Boulevard,
MC 256-80 Pasadena,
CA 91125 USA

acapponi@cs.caltech.edu
pilotto@cs.caltech.edu

G. Golino, A. Farina
Selex SI
Via Tiburtina, km12,400
00131 Rome
Italy

afarina@selex-si.com
ggolino@selex-si.com

L. Kaplan¹
U.S. Army Research
Laboratory
2800 Powder Mill Road,
Adelphi, MD 20783.

lkaplan@ieee.org

Abstract – *This paper compares two different approaches for sensor selection for distributed tracking: 1) The Frisbee method, and 2) Global Node Selection (GNS). The Frisbee method is based on the proximity of the nodes to the predicted location of the target; GNS is based on minimizing the unbiased Cramer Rao lower bound (CRLB). Both theoretical and experimental results indicate that the Frisbee method is as effective as GNS. Furthermore, the Frisbee method is attractive due to its very light computational load.*

Keywords: wireless network of sensors, surveillance, distributed target tracking.

1 Introduction

Many wireless sensor networks consist of large number of devices with limited sensing, computation and communication capabilities. These are generally battery powered and must be active only when it is necessary, to avoid inefficient energy consumption with drawback in the network lifetime. These networks are then characterised by low prime power, low data rates and small complexity (cost), while meeting the requirements for reliability and security.

For wireless sensor networks to operate effectively over an extended amount of time, it is important for the network to determine which sensors should collect and communicate measurements at any given point in time. The Frisbee method, previously proposed in [1], provides a simple means to select nodes for wireless networks consisting of binary sensors that either detect or not detect a target. In [2], the Global Node Selection (GNS) method selects the nodes based on the Cramer Rao Lower Bound

(CRLB). GNS was designed for bearings-only sensors. This paper compares the performance of the Frisbee method with GNS for binary sensors. The results indicate that the Frisbee method is always as effective as GNS. Furthermore the Frisbee method is computationally simpler than GNS.

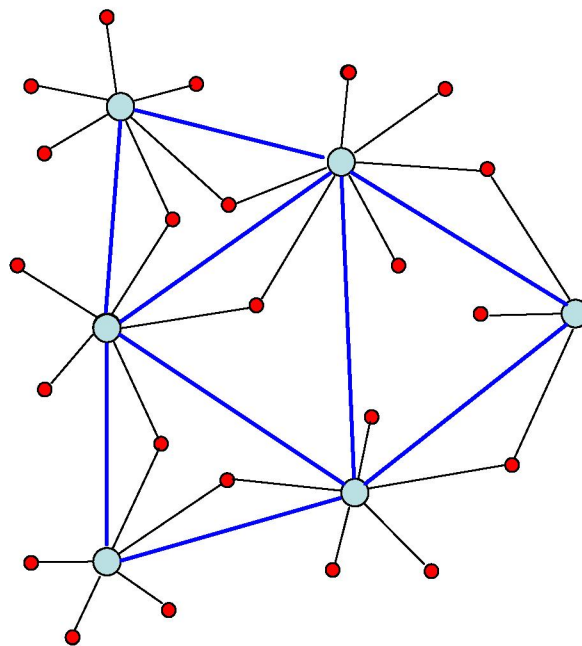


Figure 1. Typical network topology (small circle: simple sensor; large circle: complex sensor).

This paper is organized as follows. Section 2 describes the sensor network and the distributed tracker, which includes node selection. The mathematical models and foundations for node selection are detailed in Section 3.

¹ The views and conclusions contained in this document are those of the authors and should not be interpreted as presenting the official policies either express or implied of the Army Research Laboratory or the U. S. Government.

Section 4 describes different node selection approaches. Theoretical and experimental comparisons of node selection approaches are provided in Sections 5 and 6, respectively. Finally, Section 7 provides some concluding remarks.

2 Sensor networks and tracking

The wireless sensor network includes a heterogeneous set of sensors. As described in a previous article [1], we consider a network consisting of two kinds of sensors:

- simple sensors, which are able to reveal the presence of the target in their coverage area and to communicate it to complex sensors;
- complex sensors, that besides target detection and communication capability, can also activate other sensors (both simple or complex), process the data received from them, and track the targets.

This network typically has a cluster topology, where complex sensors are at the centre of a local star sub-network (see Figure 1). The network is consistent with the Zigbee standard [3].

A flow chart of the distributed tracking algorithm presented in [1] is shown in Figure 2:

- first the complex sensor selects the active simple sensors by using the predicted target position and by following a specified criterion as detailed in Section 4;
- it receives the measurements from these sensors;
- by using such measurements and the knowledge of active sensor positions, the complex sensor constructs a likelihood function;
- the maximisation of this function gives the estimation of the target position;
- the complex sensor also calculates the reliability of the estimation by computing the CRLB;
- the sensor measurement is then used to correct the prediction of the tracking algorithm;
- the algorithm finally performs the track prediction at the next step.

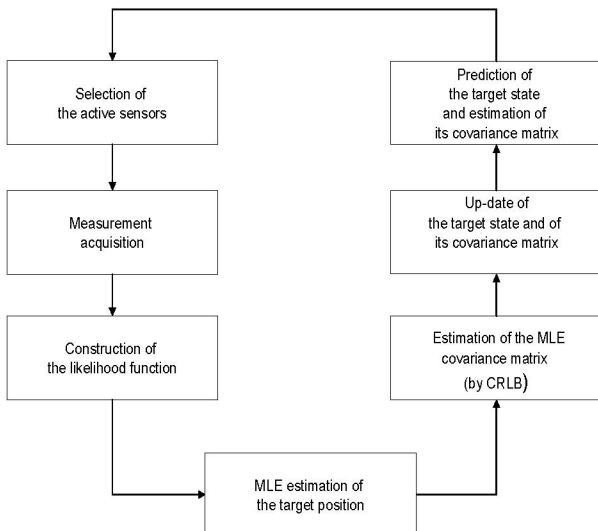


Figure 2. Flow cart of the tracking algorithm.

3 Mathematical model of the network

The simple and complex sensors are able to detect the target. Individually, they are unable to localize the range or bearing of the target. Collectively, they can communicate their binary detection decision to a complex sensor that can localize the target. The performance of any localization technique depends on the sensor model. To describe the model, let's first set up the geometry. The sensor network is composed of N_s nodes (each node is either a simple or complex sensor). The location of the i -th node can be described in Cartesian coordinates (x_i, y_i) where without loss of generality, the target is located at the origin. Alternatively, the location of the i -th node can be represented via polar coordinates relative to the target via (r_i, θ_i) , where r_i and θ_i are the range and bearing to the target, respectively. For binary sensors, the sensor model is the probability of detection (P_d) as a function of the position of the node relative to the target. A reasonable model for P_d that accounts for the spherical wave propagation is:

$$P_d(r_i) = \exp \left(\log(P_{fa}) \left(1 + \left(\frac{\log(P_{fa})}{\log(P_{d0})} - 1 \right) \cdot \frac{R_0^2}{r_i^2} \right)^{-1} \right) \quad (1)$$

where:

- P_{fa} is the false alarm probability, and
- P_{d0} is the probability of detecting the target at a range of R_0 .

The model given by (1) represents the performance of the energy detector when interrogating Swerling I targets for one coherent interval [4].

Using the architecture in Figure 2, the complex nodes chooses N_a nodes to actively sense and communicate detection results. Let $d_i \in \{0, 1\}$ be the result of the i -th node in the active set (1 if detection occurs, 0 otherwise), and \mathbf{D} represent the agglomeration of the N_a nodes, i.e., $\mathbf{D} = (d_1, \dots, d_{N_a})$. The likelihood that the target is located at (x, y) is given by (see also [5])

$$P(\mathbf{D}; x, y) = \prod_{i=1}^{N_a} P_d(R_i(x, y))^{d_i} \cdot (1 - P_d(R_i(x, y)))^{1-d_i} \quad (2)$$

where

$$R_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2}.$$

The Maximum Likelihood Estimator (MLE) determines the target location that maximizes (2). The MLE is an attractive estimator because of its desirable asymptotic properties, unbiased and efficient. However, the MLE is not necessarily a good estimator when the number of measurements is small. A centroid estimator is an alternative localization technique that estimates the target position as the centroid of the detected nodes,

$$[\hat{x}, \hat{y}]^T = \frac{\sum_{i=1}^{N_a} d_i [x_i, y_i]^T}{\sum_{i=1}^{N_a} d_i}. \quad (3)$$

An estimation of the Mean Square Error (MSE) can be found by calculating the CRLB. The CRLB is calculated by inverting the Fisher Information Matrix (FIM):

$$CRLB_{kh} = J^{-1}_{k,h} \quad (4)$$

where J is the FIM and k and h can be referred to the x and y coordinates. The trace of the matrix in (4) is a lower bound of the MSE.

For N_a binary detectors, it can be shown that the FIM is,

$$\mathbf{J} = \sum_{i=1}^{N_a} \frac{1}{P_{d_i}(1-P_{d_i})} \begin{bmatrix} \left(\frac{dP_{d_i}}{dx}\right)^2 & \frac{dP_{d_i}}{dx} \cdot \frac{dP_{d_i}}{dy} \\ \frac{dP_{d_i}}{dx} \cdot \frac{dP_{d_i}}{dy} & \left(\frac{dP_{d_i}}{dy}\right)^2 \end{bmatrix} \quad (5)$$

where $P_{d_i} = P_d(R_i(x, y))$.

The MLE and the centroid methods are ill-posed when no sensor returns a positive detection. In essence, the estimated target location is at infinity because the sensor network results indicate that a target does not exist. To be precise, the performance of the localization algorithm should be conditioned on the case that one or more of the active nodes returns a positive detection. The probability that at least one node detects the target is

$$P_{dNetwork} = 1 - \prod_{i=1}^{N_a} (1 - P_{d_i}) \quad (6)$$

The FIM conditioned on a detection is

$$\mathbf{J}_c = \frac{1}{P_{dNetwork}} \mathbf{J} - \frac{1 - P_{dNetwork}}{P_{dNetwork}} \mathbf{v} \mathbf{v}^T \quad (7)$$

where

$$\mathbf{v} = \sum_{i=1}^{N_a} \frac{1}{1 - P_{d_i}} \left[\frac{dP_{d_i}}{dx}, \frac{dP_{d_i}}{dy} \right]^T.$$

As the number of active sensors increase, $P_{dNetwork}$ converges to one. Therefore, for a sufficiently number of active sensors, $P_{dNetwork}$ is approximately one, and $\mathbf{J}_c \approx \mathbf{J}$.

It can be shown that the performance of the MLE and centroid methods actually outperform the unbiased CRLB

because they are biased. When N_a is reasonably small, it is possible to enumerate all possible 2^{N_a} detection results as D_j for $j=0, \dots, 2^{N_a}-1$, where D_0 represents the case that all nodes do not detect a target. In general, $D_j = \{d_{j0}, \dots, d_{ji}, \dots, d_{jN_a}\}$. The probability that D_j occurs conditioned on the case that $j > 0$ is

$$P_j = \frac{1}{P_{dNetwork}} \prod_{i=1}^{N_a} (P_{d_i})^{d_{ji}} (1 - P_{d_i})^{1-d_{ji}}. \quad (8)$$

Finally, the expected MS position error of any estimator $[\hat{x}(D_j), \hat{y}(D_j)]^T$ is

$$e = \sum_{j=1}^{2^{N_a}-1} P_j \left\| [\hat{x}(D_j), \hat{y}(D_j)]^T \right\|^2. \quad (9)$$

where D_j is the j -th realization of the random variable D .

4 Node Selection

4.1 Closest nodes

The tracker simply selects the N_a nodes closest to the predicted target locations. The parameter N_a allows one to trade-off localization performance with energy usage. In the case of estimation based on multilateration [6], for example, the condition of proximity is not sufficient: the sensors should surround the target over 360° . Similarly, for triangulation, the sensors also should surround the target. As shown in [2] for triangulation, the “closest” selection approach can be as effective as other selection approaches when N_a becomes larger than three. This is due to the fact that the randomly distributed nodes tend to naturally provide the necessary viewing geometry if enough nodes are employed.

4.2 Frisbee

The “Frisbee” method is also a simple approach that was originally described in [1]. For this approach, a circle is centred on the prediction of the target position and all the sensors inside it become active. To improve the stability of the algorithm, the sensors inside the previous Frisbee remain active too (“Frisbee with memory method”). Given the random distribution of sensors, this method is effective if the number of active sensors N_a is sufficiently large. The “closest” and “Frisbee” methods are similar in spirit. The primary difference is that the Frisbee approach is parameterized by a circular radius R_c rather than N_a . In fact in the “Frisbee” method the number of active sensors is not fixed as in the “closest”, but it can vary for each snapshot. For uniformly distributed nodes, the value of N_a follows a Poisson distribution with a mean value of $\lambda \pi R_c^2$, where λ is the sensor density.

4.3 Global node selection

Global node selection (GNS) chooses the N_a sensors that lead to the smallest *posterior* MS position error [2]. This paper considers a simplified version of GNS which uses the trace of the FIM inverse of the current set of measurements. This quantity is estimated by using the predicted target location.

The search for the N_a best nodes out of a field of N_s sensors is *NP-hard*. A computationally efficient implementation of GNS uses a Greedy search. As shown in [2], the Greedy search finds a nearly optimal solution. To describe the search, let's define two complementary sets of sensors:

$A := \text{set of active sensors}$

$B := \text{set of sleeping sensors}$

$$A \cup B = S = \{s_i\}_{i=1}^{N_s}$$

At the beginning, A is empty, while B consists of all the sensors in the network.

$$A = \{ \}$$

$$B = S$$

The first step is to compute the utility function, i.e., unbiased CRLB, for all the possible pairs of sensors. The pair with the best result is selected and moved from B to A :

$$A := \{s_h, s_k\}$$

$$B := S \setminus \{s_h\} \setminus \{s_k\}$$

In the successive steps, one sensor node at time is added in such a way to maximise the utility of the set A .

$$\{s_k\} / k \equiv \arg \max_i U(A \cup \{s_i \in B\})$$

$$A := A \cup \{s_k\}$$

$$B := B \setminus \{s_k\}$$

The procedure is iterated until the cardinality of A reaches the desired value of N_a .

4.4 Optimal selection

The tracker in Figure 2 uses the FIM inverse to represent the MS localization error. However, the localization methods for binary sensors are biased and their corresponding “biased” CRLBs are lower than the unbiased CRLB. In theory, the tracker can employ the expected MSE (see (9)) as the utility to optimize. We refer to such an approach as “optimal” selection. The word “optimal” is used with some caveats; namely, the approach is optimal for a specified localization method

when the predicted target location matches the actual target location. Unlike the calculation of the FIM, which requires $O(N_a)$ computations for a given hypothesized active set, the calculation requires $O(2^{N_a})$ operations. Even the Greedy search described in Section 4.3 does not make optimal selection feasible in practice. This paper uses the optimal approach as a means to baseline other approaches in Section 5. Furthermore, the calculation of (9) using the MLE is expensive because of the required search algorithm. Therefore, the optimal method is computed for centroid method.

5 Clairvoyant analysis

This section compares the “closest” and GNS selection approaches for binary sensors relative to the optimal method under the condition that the true target location is used instead of the predicted target location. This clairvoyant comparison provides useful insights about the performance of node selection approaches under controlled conditions. For these tests, the track filter does not need to be employed. Since the processing is all offline, the GNS minimizes the unbiased CRLB via exhaustive search, rather than the Greedy search.

To evaluate the approaches, we generated 1000 network configurations consisting of $N_s=15$ nodes by randomly placing the nodes via the uniform distribution within a circular region of radius 100m around the true target location. The sensor observation model follows (1) where $P_{fa} = 0.1$ and $P_{d0} = 78\%$ or 28% for $R_0 = 3.5\text{m}$. For each configuration, the best active set chosen by the “closest” and GNS methods are compared against the set chosen by the optimal approach, where the centroid method is the localization method. The values for N_a ranged from two to five.

Figure 3 shows the percentage of configurations where the closest and GNS method selected the same active set as the optimal approach, i.e., the set that corresponds to the lowest MSE at the target location. Figure 4 plots the average relative increase in RMS position error with respect to the optimal set. The closest set usually corresponds to the optimal set, and the relative increase in error is less than 1%. The GNS-selected set actually leads to higher error, but the relative increase in errors levels out to a modest 16% as N_a reaches four. Finally, the relative performance of the closest and Fisher do not appear to be sensitive to the quality of the sensor, i.e., P_{d0} .

The clairvoyant analysis indicates that even when N_a is small, the proximity of the sensors to the target is the most important feature for good localization when employing binary sensors. Apparently, the biases of binary sensor localization limit the usefulness of the Fisher-based metric used by the GNS method (the trace of the inverse of the FIM), which enforces the active set to surround the target. As N_a becomes large, the closest set of nodes to the target will naturally surround the target. Therefore, one would expect that selection method such as the “Frisbee” and “closest” methods should be as effective as GNS for any value of N_a .

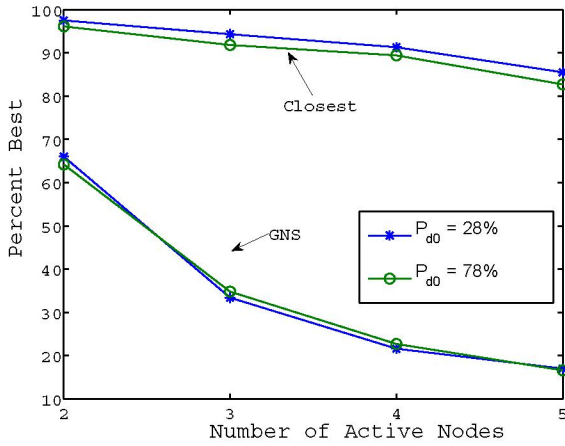


Figure 3: Probability that node selected via “closest” or GNS methods correspond to the best set.

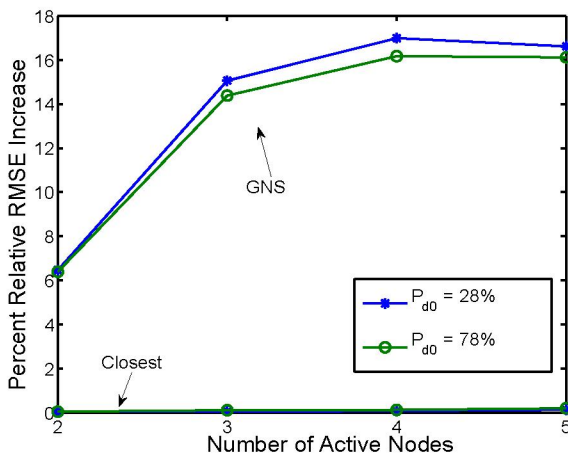


Figure 4: Relative increase in the RMS error with respect to the optimal set when selecting nodes via the closest and GNS approaches.

6 Tracking analysis

This section illustrates the overall performance of the tracker in Figure 2 using either the “Frisbee” or GNS approaches for node selection. The performance numbers are obtained through software that runs Monte Carlo simulations of a random network employing target detection and tracking [1]. The tracker consists of a linear Kalman filter using a four state white noise acceleration model where the state vector includes the x and y coordinates and their derivatives. The MLE algorithm is used to estimate the target position from the detection data of the active sensors, and the CRLB is used to represent the measurement error covariance matrix. To calculation of the CRLB assumes that at least one sensor per scan detects the target ($P_{d,Network} \approx 1$). This assumption is reasonable for the values of active sensors and Frisbee radius fixed for the simulation. Section 6.1 details the scenario, and Section 6.2 provides the performance results.

6.1 The tracking scenario

Targets are detected and tracked within a square surveillance area by randomly dispersing $N_s=100$ sensors. The resulting sensor network consists of 80 simple and 20 complex nodes. The size of the surveillance area is 20m by 20m in the example; but if it were larger, e.g., 20km by 20km, the corresponding MS tracking error will increase accordingly. Each sensor is placed in the surveillance area according to a uniform distribution. Then, the sensors are randomly shifted by a Gaussian-distributed displacement to simulate the inaccuracy in the placing process. The standard deviation of the displacement is 0.2 m. The location of each sensor is supposed to be known by the network; e.g., it can be determined by a network initialization process (i.e. by performing multilateration [6]). The 80% P_d detection range for each sensor is 3.5 m. The connectivity of the network is guaranteed as described in [7].

For most collection snapshots, the target is moving through the surveillance area with a constant velocity. About halfway through the region, the target makes a single manoeuvre and changes direction. The stars in Figure 3 show the position of the target at each snapshot, or data collection, interval. The number of snapshot intervals is 49.

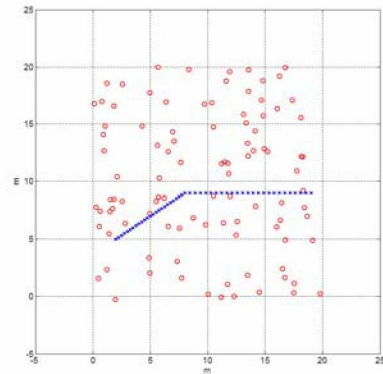


Figure 5. Target trajectory (crosses) and sensor positions (circles).

6.2 Simulation results

We ran simulations to measure the overall estimation performance of the tracker when employing either the Frisbee method for three values of the Frisbee radius ($R_c=5, 6, 7$ m) or the GNS using three values for the numbers of active sensors ($N_a=16, 24, 32$). Figure 6 shows the average number of sensors activated by the Frisbee of different radius. The number of active sensors is bigger near the centre of the surveillance area, while it is smaller near the border.

Figures 7, 9, 11 and 13 plot the statistics obtained by using the GNS method, i.e., bias and standard deviation of the Cartesian coordinates, as function of the snapshot interval. Similarly, Figures 8, 10, 12 and 14 provide the

localization statistics for the Frisbee method. These figures show no meaningful difference between the performance of the GNS and Frisbee methods.

Figures 15 and 17 highlight which sensors the GNS with $N_a=16$ activate during the 10th and 35th snapshot interval. Similarly, Figures 16 and 18 illustrate the location of the active sensor via the Frisbee method using $R_c=5\text{m}$ for the same intervals. The figures show that the majority of the active sensors selected by the two algorithms are coincident. Similar to the Frisbee method, the GNS also tends to activate the sensors that are close to the target position predicted by the tracking algorithm when N_a is large. This result is not surprising for a large number of active sensors. Furthermore, N_a can not be too small, because the binary sensors will lead to a poor localization. Even if N_a could be small, the clairvoyant results from the previous section indicate that the Frisbee method would still be as effective as GNS.

7 Conclusions

This paper compares two different algorithms for sensor selection in wireless network distributed tracking where the sensors are simple detectors. Namely, the sensors can only communicate whether or not they detect the presence of a target. The sensor selection methods under test include the Frisbee method, proximity based, and GNS, which balances proximity and distribution around the target. The theoretical results were evaluated for small values of N_a where the selection methods employed the true target locations. The experimental results evaluated the selection methods under the more realistic case that they use the predicted target locations. Both theoretical and experimental results indicate that the Frisbee is as effective as GNS for any number of active sensors N_a . These results are different than the case of bearings-only sensors where GNS was shown to outperform a proximity based selection approach when N_a is in the order of two to three nodes per snapshot [2]. Because of its computational simplicity, the Frisbee method is an appealing choice for surveillance networks consisting of binary sensors.

Future work will investigate simple methods to further conserve energy. For instance, the simple sensors can employ a random protocol to enter into awake and sleep cycles. The protocol should be developed to ensure that a sufficient number of nodes are available inside a Frisbee so that a desired localization performance can be obtained. Furthermore, the density of awake nodes should be large enough to ensure connectivity of the nodes for message passing. Some initial work in [7] will be helpful to the development of such protocols.

References

- [1] A. Capponi, A. Farina, G. Golino, C. Pilotto, "Surveillance by means of a random sensor network: a heterogeneous sensor approach", Proceedings of 8th International Conference on Information Fusion, Philadelphia, USA, July 2005.
- [2] L. M. Kaplan, "Global node selection for localization in a distributed network of bearing-only sensors", IEEE Transactions on Aerospace and Electronic Systems, Vol. 42, No. 1, pp. 113-135, Jan. 2006.
- [3] Homepage of ZigbeeTM alliance www.zigbee.org.
- [4] S. Blackman and R. Popoli, Design and analysis of modern tracking systems, Norwood, MA: Artech House, 1999.
- [5] A.A. Rodriguez, M. Lazaro, L. Tong, "Target local estimation in sensor networks using range information", IEEE Sensor Array and Multichannel Signal Processing workshop, 19-21 July 2004, Sitges, Spain.
- [6] N. Patwari, J. A. Nash, S. Kyperountas, A. O. Hero III, R. L. Moses, N. S. Correal, "Locating the nodes", IEEE SP Magazine, No. 54, July 2005.
- [7] A. Capponi, A. Farina, G. Golino, L. Kaplan, C. Pilotto, "Connectivity for the Frisbee architecture", submitted to the Proceedings of 9th International Conference on Information Fusion, Florence, Italy, June 2006.

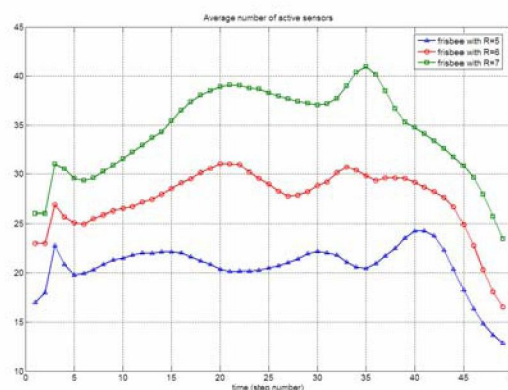


Figure 6. Average number of active sensors along the trajectory (Frisbee method).

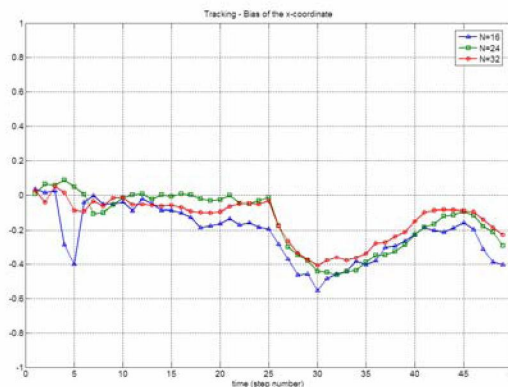


Figure 7. Bias of the x-coordinate estimation, GNS method. Results averaged over 500 Monte Carlo trials.

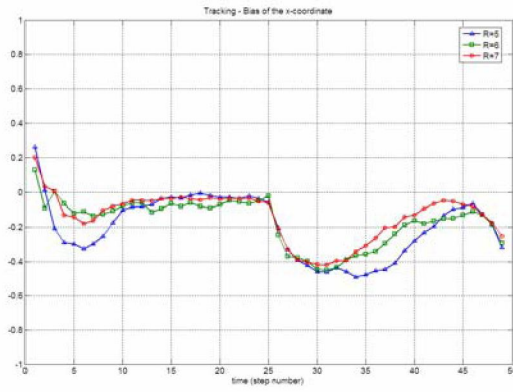


Figure 8. Bias of the x-coordinate estimation, Frisbee method. Results averaged over 500 Monte Carlo trials.

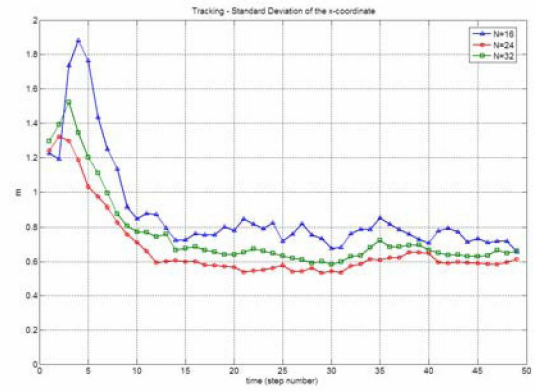


Figure 11. Standard deviation of the x-coordinate estimation, GNS method. Results averaged over 500 Monte Carlo trials.

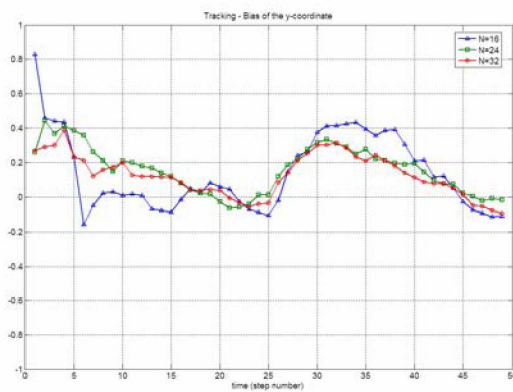


Figure 9. Bias of the y-coordinate estimation, GNS method. Results averaged over 500 Monte Carlo trials.

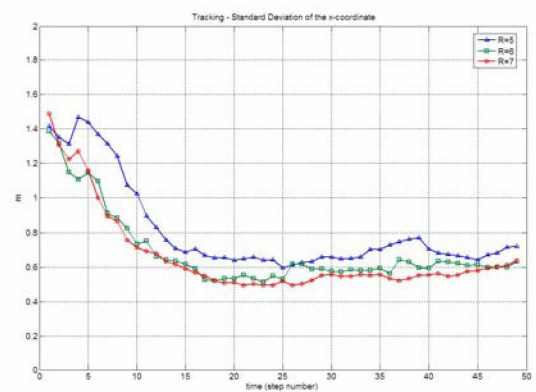


Figure 12. Standard deviation of the x-coordinate estimation, Frisbee method. Results averaged over 500 Monte Carlo trials.

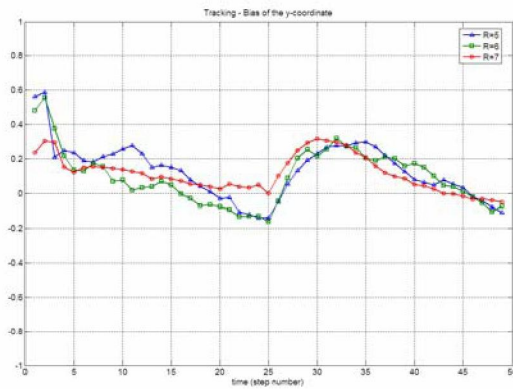


Figure 10. Bias of the y-coordinate estimation, Frisbee method. Results averaged over 500 Monte Carlo trials.

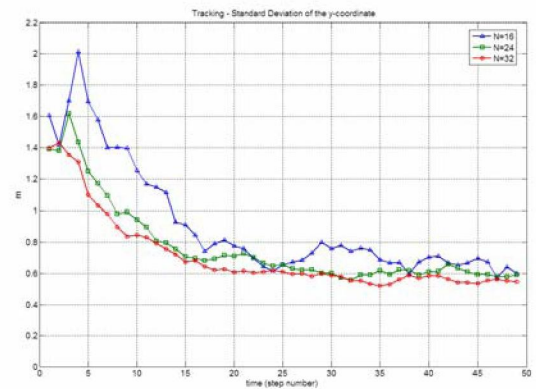


Figure 13. Standard deviation of the y-coordinate estimation, GNS method. Results averaged over 500 Monte Carlo trials.

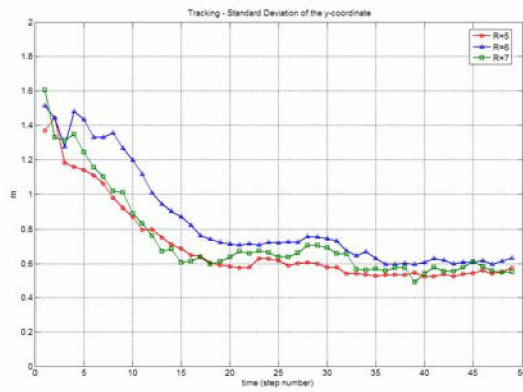


Figure 14. Standard deviation of the y-coordinate estimation, Frisbee method. Results averaged over 500 Monte Carlo trials.

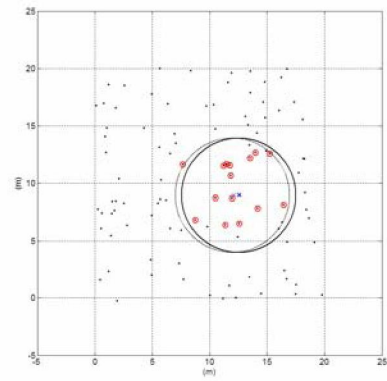


Figure 17. Sensors activated by the GNS with $M=16$ in more than 50% of the trials; 35th step of the tracking algorithm.

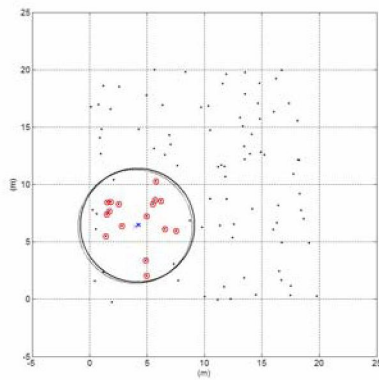


Figure 15. Sensors activated by the GNS with $M=16$ in more than 50% of the trials; 10th step of the tracking algorithm.

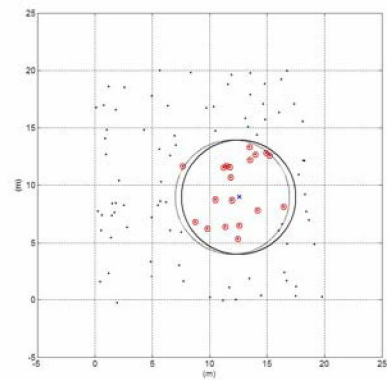


Figure 18. Sensors activated by the Frisbee with $R=5$ in more than 50% of the trials; 35th step of the tracking algorithm.

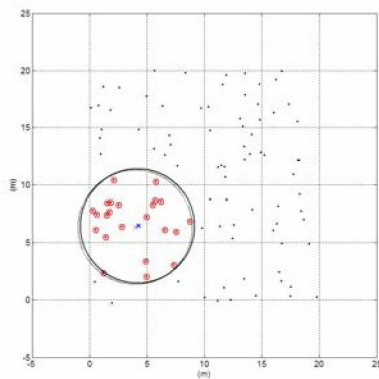


Figure 16. Sensors activated by the Frisbee with $R=5$ in more than 50% of the trials; 10th step of the tracking algorithm.